Machete Cálculo Lambda

1 Tipos y términos

Las expresiones de tipos (o simplemente tipos) son

$$\sigma ::= Bool \mid Nat \mid \sigma \to \sigma$$

Sea $\mathcal{X}$ un conjunto infinito enumerable de variables y $x \in \mathcal{X}$. Los términos están dados por

$$\begin{aligned} M ::=& x \\ &\mid true \\ &\mid false \\ &\mid if\ M\ then\ M\ else\ M \\ &\mid \lambda x : \sigma.M \\ &\mid MM \\ &\mid 0 \\ &\mid succ(M) \\ &\mid pred(M) \\ &\mid iszero(M) \end{aligned}$$

2 Axiomas y Reglas de Tipado

Variables.

$$\frac{x : \sigma \in \Gamma}{\Gamma \vdash x : \sigma}(\text{T-Var})$$

Bool.

$$\frac{}{\Gamma \vdash true : Bool}(\text{T-True}) \qquad \frac{}{\Gamma \vdash false : Bool}(\text{T-False})$$

$$\frac{\Gamma \vdash M : Bool \quad \Gamma \vdash P : \sigma \quad \Gamma \vdash Q : \sigma}{\Gamma \vdash if\ M\ then\ P\ else\ Q : \sigma}(\text{T-If})$$

Abstracción.

$$\frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}(\text{T-App}) \qquad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x : \sigma. M : \sigma \to \tau}(\text{T-Abs})$$

Naturales

$$\frac{}{\Gamma \vdash 0 : Nat}(\text{T-Zero})$$

$$\frac{\Gamma \vdash M : Nat}{\Gamma \vdash succ(M) : Nat}(\text{T-Succ}) \qquad \frac{\Gamma \vdash M : Nat}{\Gamma \vdash pred(M) : Nat}(\text{T-Pred}) \qquad \frac{\Gamma \vdash M : Nat}{\Gamma \vdash iszero(M) : Bool}(\text{T-IsZero})$$

3 Semántica Operacional

$$V ::= true \mid false \mid \lambda x : \sigma. M \mid succ(V)$$

Los valores de tipo Nat pueden escribirse como $\underline{n}$, lo cual abrevia $succ^n(0)$.

Aplicación.

$$\frac{M_1 \to M_1'}{M_1 M_2 \to M_1' M_2}(\text{E-App1}) \qquad \frac{M_2 \to M_2'}{V_1 M_2 \to V_1 M_2'}(\text{E-App2})$$

$$\frac{}{(\lambda x : \sigma. M)V \to M\{x \leftarrow V\}}(\text{E-AppAbs})$$

IfThenElse.

$$\frac{M_1 \to M_1'}{if\ M_1\ then\ M_2\ else\ M_3 \to if\ M_1'\ then\ M2\ else\ M3}(\text{E-If})$$

$$\frac{}{if\ true\ then\ M_2\ else\ M_3 \to M_2}(\text{E-IfTrue}) \qquad \frac{}{if\ false\ then\ M_2\ else\ M_3 \to M_3}(\text{E-IfFalse})$$

Sucesor.

$$\frac{M_1 \to M_1'}{succ(M_1) \to succ(M_1')}(\text{E-Succ})$$

Predecesor.

$$\frac{M_1 \to M_1'}{pred(M_1) \to pred(M_1')}(\text{E-Pred})$$

$$\frac{}{pred(0) \to 0}(\text{E-PredZero}) \qquad \frac{}{pred(succ(\underline{n})) \to \underline{n}}(\text{E-PredSucc})$$

IsZero.

$$\frac{M_1 \to M_1'}{iszero(M_1) \to iszero(M_1')}(\text{E-IsZero})$$

$$\frac{}{iszero(0) \to true}(\text{E-IsZeroZero}) \qquad \frac{}{iszero(succ(\underline{n})) \to false}(\text{E-IsZeroSucc})$$

4 Extensión Unit

Unit es un tipo unitario y el único valor posible de una expresión de ese tipo es *unit*. Cumple un rol similar a *void* en C o Java.

$$\sigma ::= \ldots \mid Unit$$

$$M ::= \ldots \mid unit$$

Agregamos el axioma de tipado:

$$\frac{}{\Gamma \vdash unit : Unit}(\text{T-U}\textsc{nit})$$

Notemos que no hay reglas de evaluación nuevas. Extendemos el conjunto de valores V con *unit*. Su utilidad principal es en lenguajes con efectos laterales.

$$V ::= \ldots \mid unit$$

5 Extensión Registros

$$\sigma ::= \ldots \mid \{l_i : \sigma_i^{i\in 1..n}$$

$$M ::= \ldots \mid \{l_i = M_i^{i\in 1..n} \mid M.l$$

Agregamos las siguientes reglas de tipado:

$$\frac{\Gamma \vdash M_i : \sigma_i \quad \text{para cada } i \in 1..n}{\Gamma \vdash \{l_i = M_i{}^{i\in 1..n}\} : \{l_i : \sigma_i{}^{i\in 1..n}\}}(\text{T-R}\textsc{cd})$$

$$\frac{\Gamma \vdash M : \{l_i : \sigma_i{}^{i\in 1..n}\} \qquad j \in 1..n}{\Gamma \vdash M.l_j : \sigma_j}(\text{T-P}\textsc{roj})$$

Extendemos el conjunto de valores V de la siguiente manera:

$$V ::= \ldots \mid \{l_i = V_i{}^{i\in 1..n}\}$$

Agregamos las siguientes reglas de evaluación:

$$\frac{M_j \to M_j'}{\{l_i = \mathbf{V_i}{}^{i\in 1..j-1}, \mathbf{l_j} = \mathbf{M_j}, l_i = M_i{}^{i\in j+1,..n}\} \to \{l_i = \mathbf{V_i}{}^{i\in 1..j-1}, \mathbf{l_j} = \mathbf{M_j'}, l_i = M_i{}^{i\in j+1,..n}\}}(\text{E-R}\textsc{cd})$$

$$\frac{j \in 1..n}{\{l_i = V_i{}^{i\in 1..n}\}.l_j \to V_j}(\text{E-P}\textsc{roj}\text{R}\textsc{cd}) \qquad \frac{M \to M'}{M.l \to M'.l}(\text{E-P}\textsc{roj})$$

6 Extensión Let

$$M ::= \ldots \mid let\ x : \sigma = M\ in\ N$$

Descripción informal. $let\ x : \sigma = M\ in\ N$ nos permite evaluar a M hasta obtener un valor V, que luego vamos a ligar a x y, finalmente, evaluar N reemplazando a x por V. Notemos que no estamos agregando tipos nuevos. El valor de x permanece inmutable a lo largo de la evaluación de N.

Agregamos las siguientes reglas de tipado:

$$\frac{\Gamma \vdash M : \sigma_1 \qquad \Gamma, x : \sigma_1 \vdash N : \sigma_2}{\Gamma \vdash let\ x : \sigma_1 = M\ in\ N : \sigma_2}(\text{T-L}\textsc{et})$$

Agregamos las siguientes reglas de evaluación:

$$\frac{M_1 \to M_1'}{let\ x : \sigma = M_1\ in\ M_2 \to let\ x : \sigma = M_1'\ in\ M_2}(\text{E-L}\textsc{et})$$

$$\frac{}{let\ x : \sigma = V_1\ in\ M_2 \to M\{x \leftarrow V_1\}}(\text{E-L}\textsc{et}\text{V})$$

7 Extensión Fix

Introducimos las ecuaciones recursivas con el operador de punto fijo **fix**.

$$M ::= \dots \mid fix\ M$$

No se precisan nuevos tipos, pero sí una regla de tipado.

$$\frac{\Gamma \vdash M : \sigma_1 \to \sigma_1}{\Gamma \vdash fix\ M : \sigma_1}(\text{T-Fix})$$

No hay valores nuevos, pero sí reglas de evaluación nuevas.

$$\frac{M_1 \to M_1'}{fix\ M_1 \to fix\ M_1'}(\text{E-Fix})$$

$$\frac{}{fix\ (\lambda x : \sigma. M) \to M\{x \leftarrow fix\ (\lambda x : \sigma. M)\}}(\text{E-FixBeta})$$

Link al documento https://www.overleaf.com/8678714595mhfmttskkhdc.